Discrete-time Temporal Reasoning with Horn DLRs

Peter Jonsson[*] **& Tomas Lööw**[†]
Department of Computer and Information Science
Linköping University
peter.jonsson@liu.se, tomas.andersson@liu.se

Abstract

Temporal reasoning problems arise in many areas of AI, including planning, natural language understanding, and reasoning about physical systems. The computational complexity of continuous-time temporal constraint reasoning is fairly well understood. There are, however, many different cases where discrete time must be considered; various scheduling problems and reasoning about sampled physical systems are two examples. Here, the complexity of temporal reasoning is not as well-studied nor as well-understood. In order to get a better understanding, we consider the powerful Horn DLR formalism adapted for discrete time and study its computational complexity. We show that the full formalism is NP-hard and identify several maximal tractable subclasses. We also 'lift' the maximality results to obtain hardness results for other families of constraints. Finally, we discuss how the results and techniques presented in this paper can be used for studying even more expressive classes of temporal constraints.

1 Introduction

Reasoning about time is ubiquitous in artificial intelligence and many different branches of computer science. Noteworthy examples include planning, diagnosis, and temporal databases. For a general overview of temporal reasoning, see, for instance, the handbook [Fisher *et al.*, 2005]. The temporal constraint satisfaction problem is very well-studied and there has lately been substantial progress in understanding the complexity of this problem. Bodirsky and Kára [2010] have presented a complete classification of the temporal constraint problem for relations that are first-order definable in the structure $(\mathbb{Q}; <)$. This result subsumes much of the previous work on *qualitative* (that is, the case where we cannot refer to individual time points in the underlying time structure) temporal constraints such as Allen's algebra. There are no such unifying result for metric temporal constraints, but many partial results are known, cf. [Jonsson and Bäckström, 1998; Krokhin *et al.*, 2004].

The situation is very different if we turn our attention to *discrete* temporal constraints where the set of time points is some subset of the set of integers $\mathbb{Z}$. There are some scattered complexity results (cf. [Bettini *et al.*, 1998; Meiri, 1996]) but a coherent picture is lacking. This is unsatisfactory since reasoning about discrete time is an important part of AI: let us just mention temporal logics, plan generation, and discrete time Markov chains as three concrete examples. Reasoning about discrete time is also inevitable in many 'industrial' settings: for systems that are repeatedly sampled (for monitoring or other purposes), we are implicitly forced to assume that the underlying model of time is discrete. Our goal with this paper is to initiate a systematic study of temporal constraint satisfaction under the assumption that time is discrete instead of continuous. The focus will be on the computational complexity of such problems; more precisely, we aim at identifying restricted classes of constraints such that the corresponding constraint satisfaction problem can be solved in polynomial time. Obtaining a full classification of hard and easy cases is of course highly desirable — it gives us a very powerful tool for studying the complexity of problems that can be modelled within the language. Since temporal constraint reasoning appears as a subproblem in many different types of automated reasoning, we expect such results to be useful in many other contexts, too. For instance, note that discrete semilinear relations (to be defined later on) have been used intensively for a long time in, for example, formal verification [Boujjani and Habermehl, 1996], distributed computing [Angluin *et al.*, 2007], and automata therory [Parikh, 1966],

In order to introduce temporal constraint reasoning formally, we first define the general constraint satisfaction problem.

Definition 1 Let Γ be a set of finitary relations over some set D of values. The constraint satisfaction problem over Γ (CSP(Γ)) is defined as follows:

Instance: A set V of variables and a set C of constraint applications $R(v_1, \ldots, v_k)$ where k is the arity of R, $v_1, \ldots, v_k \in V$ and $R \in \Gamma$.
Question: Is there a total function $f : V \to D$ such that $(f(v_1), \ldots, f(v_k)) \in R$ for each constraint $R(v_1, \ldots, v_k)$ in

[*]Partially supported by the *Swedish Research Council* (VR) under grant 621-2009-4431

[†]Partially supported by the *Swedish National Graduate School in Computer Science* (CUGS).

C?

The set Γ is referred to as the *constraint language*. Given a set D, we let $\Gamma|_D$ denote Γ restricted to D, i.e. $\Gamma|_D = \{R \cap D^n \mid R \in \Gamma \text{ and } R \text{ has arity } n\}$. We sometimes slightly abuse notation to avoid unnecessary clutter. For instance, we may say 'the relation $x = y + z$' instead of 'the relation $\{(x, y, z) \in \mathbb{Z}^3 \mid x = y + z\}$.' A constraint satisfaction problem CSP(Γ) is *globally tractable* if CSP(Γ) is in P and *locally tractable* if CSP(Γ') is in P for every finite set $\Gamma' \subseteq \Gamma$. Similarly, CSP(Γ) is *globally NP-hard* if CSP(Γ) is NP-hard and *locally NP-hard* if CSP(Γ') is NP-hard for some finite set $\Gamma' \subseteq \Gamma$.

The separation of local and global tractability/NP-hardness is motivated by the following result: let $\langle\Gamma\rangle$ (the *closure* or *co-clone* of Γ) denote all relations that are *pp-definable* in Γ. A relation R is pp-definable in Γ if it can be defined by a first-order formula over Γ without using disjunction and negation, and with only existential quantification.

Theorem 2 [Jeavons, 1998] For every finite $\Theta \subseteq \langle\Gamma\rangle$, CSP($\Theta$) is polynomial-time reducible to CSP(Γ). Furthermore, if $R \in \langle\Gamma\rangle$, then CSP($\Gamma \cup \{R\}$) and CSP($\Gamma$) are polynomial-time equivalent problems.

This implies, for instance, that if CSP(Γ) is globally tractable, then CSP($\langle\Gamma\rangle$) is locally tractable.

Let us now turn our attention to temporal constraint problems. We let $D \subseteq \mathbb{R}$ denote a set of *time points*. Let the set $\mathcal{S}_D$ contain all relations $\{(x_1, \ldots, x_n) \in D^n \mid C_1 \wedge \ldots \wedge C_k\}$ where each clause C_i denotes a disjunction $(p_1 r_1 c_1 \vee \ldots \vee p_m r_m c_m)$. Here, c_j is an integer, $r_j \in \{<, \leq, =, \neq, \geq, >\}$ and $p_j(x_1, \ldots, x_n)$ is a linear polynomial (i.e. the degree of p equals one) with integer coefficients. We adopt a simple representation of relations in $\mathcal{S}_D$: every relation R in $\mathcal{S}_D$ is represented by its defining formula where each coefficient is written in binary. Let $\mathcal{D}_D \subseteq \mathcal{S}_D$ contain the relations that are defined by a single clause. Let $\mathcal{H}_D \subseteq \mathcal{D}_D$ contain the relations that are defined by a single clause that contains at most one relation that is not of the type $p(\bar{x}) \neq c$. The names $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{H}$ are chosen to reflect the names given to the corresponding relations in the literature: the relations in $\mathcal{S}_D$ are called *semilinear relations*, the relations in $\mathcal{D}_D$ are called *disjunctive linear relations* (DLRs), and the relations in $\mathcal{H}_D$ are called *Horn DLRs*. DLRs and Horn DLRs were introduced in [Jonsson and Bäckström, 1998] but only for continuous time structures (in fact, only for the set $\mathbb{R}$ of real numbers).

Before we continue, we need some NP-hardness results. For distinct $a, b \in \mathbb{Z}$, define $T_{a,b} = \{(a, a, b), (a, b, a), (b, a, a)\}$. Clearly, CSP($\{T_{a,b}\}$) is NP-hard problems since it corresponds to 1-IN-3-SAT restricted to clauses without negated literals.

Theorem 3 CSP($\mathcal{H}_\mathbb{R}$) is globally tractable while CSP($\mathcal{H}_\mathbb{Z}$) is locally NP-hard. Furthermore, CSP($\mathcal{D}_D$) and CSP($\mathcal{S}_D$) are locally NP-hard when $D \in \{\mathbb{Z}, \mathbb{R}\}$.

Proof: Global tractability of CSP($\mathcal{H}_\mathbb{R}$) and local NP-hardness of CSP($\mathcal{D}_\mathbb{R}$) and CSP($\mathcal{S}_\mathbb{R}$) follows from [Jonsson and Bäckström, 1998]. For the remaining cases, it is sufficient to prove local NP-hardness of CSP($\mathcal{H}_\mathbb{Z}$). Simply note that we can pp-define $T_{0,1}$ in $\mathcal{H}_\mathbb{Z}$ by $T_{0,1}(x, y, z) \equiv x \geq 0 \wedge y \geq 0 \wedge z \geq 0 \wedge x + y + z = 1$ and apply Theorem 2. $\square$

Since CSP($\mathcal{H}_\mathbb{Z}$) is locally NP-hard, it makes sense to start looking for tractable fragments within $\mathcal{H}_\mathbb{Z}$, and this is a natural first step in a bottom-up approach to classifying the complexity of CSP($\mathcal{D}_\mathbb{Z}$) and CSP($\mathcal{S}_\mathbb{Z}$). Also note that the modelling power (in continuous time) of $\mathcal{H}_\mathbb{R}$ is quite high; many tractable fragments described in the literature are within $\mathcal{H}_\mathbb{R}$ [Jonsson and Bäckström, 1998]. This indicates that $\mathcal{H}_\mathbb{Z}$ may be interesting from a modelling point of view, too.

Due to the NP-hardness of CSP($\mathcal{H}_\mathbb{Z}$), we will concentrate on identifying tractable fragments and study their *maximality* in the forthcoming three sections. Given constraint languages $\Gamma \subseteq \Theta$, we say that Γ is *maximally tractable* in Θ if CSP(Γ) is globally tractable and CSP($\Gamma \cup \{R\}$) is locally NP-hard for every $R \in \Theta \setminus \Gamma$. Maximality can obviously be defined in different ways with respect to local and global properties but this definition is sufficent for our purposes. We consider problems where solutions can be 'scaled' in Section 2, problems connected to linear equations in Section 3, and so-called k-valid constraints in Section 4. In the proofs, we demonstrate how concepts and ideas like *reduced formulas* [Bodirsky *et al.*, 2010a] and the *independence property* [Cohen *et al.*, 2000] can be used for studying discrete-time temporal constraints. We also show how some of the maximality results can be generalised to hardness results for larger classes of constraints. We conclude the paper with a brief discussion concerning the results and future research directions.

2 Scalable constraints

One way to start looking for tractable fragments of $\mathcal{H}_\mathbb{Z}$ is to ask under which circumstances a solution to an instance I of CSP($\mathcal{H}_\mathbb{R}$) implies a solution to the corresponding instance $I|_\mathbb{Z}$ of CSP($\mathcal{H}_\mathbb{Z}$). We begin with the following lemma.

Lemma 4 Let Γ be a constraint language over $\mathbb{R}$ such that the following holds.

1. Every satisfiable instance of CSP(Γ) is satisfied by some rational point.
2. For each $R \in \Gamma$, it holds that if $\bar{x} = (x_1, x_2, \ldots, x_k) \in R$, then $(ax_1, ax_2, \ldots, ax_k) \in R$ for all $a \in \{y \in \mathbb{R} \mid y \geq 1\} \setminus X$ where X is a (possibly empty) finite set. The set X may depend on both R and $\bar{x}$.
3. CSP(Γ) is globally (or locally) tractable.

Then, the problem CSP($\Gamma|_\mathbb{Z}$) is also globally (or locally) tractable.

Proof: Let I be an arbitrary satisfiable instance of CSP(Γ) with a rational solution $\bar{x} = (x_1/y_1, \ldots, x_k/y_k)$ where $x_1, \ldots, x_k \in \mathbb{Z}$ and $y_1, \ldots, y_k \in \mathbb{Z}^+ \setminus \{0\}$. Let $n = \prod_{i=1}^{k} y_i$ and note that $n \geq 1$.

For an arbitrary constraint R in I, we know that it is satisfied by $a\bar{x}$ for every $a \in \{y \in \mathbb{R} \mid y \geq 1\} \setminus X$ where X is finite. For every constraint C_i in I, let X_i denote the set of 'exception' points, and let $t = \sum_{i=1}^{m} |X_i|$ (where m is the number of constraints in I).

It follows that there is at least one a in the set $\{y \in \mathbb{Z} \mid 1 \leq y \leq t+1\}$ such that $an\bar{x}$ satisfies I. The vector $an\bar{x}$ is integral due to choice of n which concludes the proof. $\square$

Given a real vector $\bar{x} = (x_1, \ldots, x_k)$, let $||\bar{x}||$ denote its Euclidean norm, i.e. $\sqrt{x_1^2 + \ldots + x_k^2}$. Recall that $||\bar{x}+\bar{y}|| \leq ||\bar{x}|| + ||\bar{y}||$ and $||\alpha\bar{x}|| = |\alpha| \cdot ||\bar{x}||$ for all real vectors $\bar{x}, \bar{y}$ and arbitrary $\alpha \in \mathbb{R}$.

Theorem 5 If I is a satisfiable instance of CSP($\mathcal{S}_{\mathbb{R}}$), then I is satisfied by at least one rational point.

Proof: Let $\bar{r}$ be a satisfying real point. Assume I contains the constraints $\{C_0, \ldots, C_n\}$ where each C_i is a disjunction $l_{i1} \vee l_{i2} \vee \ldots \vee l_{ik}$. There is (at least) one l_{ij} from each C_i that is satisfied by $\bar{r}$. Since $a \leq b \equiv a < b \vee a = b$, $a \geq b \equiv a > b \vee a = b$, and $a \neq b \equiv a < b \vee a > b$, we can without loss of generality assume that either $l_{ij} \equiv p(\bar{x}) < c$ or $l_{ij} \equiv p(\bar{x}) = c$. It is clearly sufficient to find a rational satisfying point, $\bar{q}$, that satisfies the formula $l_{0j_0} \wedge \ldots \wedge l_{nj_n}$.

First consider the literals of the type $p(\bar{x}) < c$. The sets of satisfying points to these kinds of relations are clearly open. Hence, there is some rational number $\delta > 0$ so that all points $\bar{x}$ for which $||\bar{r} - \bar{x}|| < \delta$ satisfy these relations.

The remaining literals are of the form $p(\bar{x}) = c$ and we can view them as a linear equation system $A\bar{x} = \bar{b}$. Every satisfiable system of linear equations has a rational solution and a vector $\bar{x}$ is a solution if and only if it can be expressed as $\bar{x} = \bar{c} + x_1\bar{v}_1 + \ldots + x_k\bar{v}_k$ where $A\bar{v}_i = \bar{0}$, $A\bar{c} = \bar{b}$, $\bar{c}, \bar{v}_1, \ldots, \bar{v}_k$ are rational vectors, and $x_1, \ldots, x_k$ are real numbers.

Since $\bar{r}$ satisfies $A\bar{r} = \bar{b}$, it can be expressed as $\bar{r} = \bar{c} + r_1\bar{v}_1 + \ldots r_k\bar{v}_k$, with $r_i \in \mathbb{R}$. The rational numbers are dense in the real numbers so we can find rational numbers q_i satisfying $|r_i - q_i| < \delta_e$ for all i and for any $\delta_e > 0$. Let $\bar{q} = \bar{c} + q_1\bar{v}_1 + \ldots + q_k\bar{v}_k$ and we find that $||\bar{r} - \bar{q}|| = ||(r_1 - q_1)\bar{v}_1 + \ldots + (r_k - q_k)\bar{v}_k|| \leq |r_1 - q_1| \cdot ||\bar{v}_1|| + \ldots + |r_k - q_k| \cdot ||\bar{v}_k|| < \delta_e \cdot (||\bar{v}_1|| + \ldots + ||\bar{v}_k||)$. By choosing $\bar{q}$ so that δ_e gets sufficiently small, we can achieve $||\bar{r} - \bar{q}|| < \delta$. It follows that $\bar{q}$ satisfies $l_{0j_0} \wedge l_{1j_1} \wedge \ldots \wedge l_{nj_n}$. $\square$

Thus, $\mathcal{H}_{\mathbb{R}}$ satisfies requirement 1) and 3) of Lemma 4. We let $\Lambda_{\mathbb{Z}} \subseteq \mathcal{H}_{\mathbb{Z}}$ contain the relations that satisfy requirement 2) and have thus proved the following.

Theorem 6 The problem, CSP($\Lambda_{\mathbb{Z}}$) is tractable.

We now verify that $\Lambda_{\mathbb{Z}}$ is maximally tractable in $\mathcal{H}_{\mathbb{Z}}$. We need the concept of *reduced relations*.

Definition 7 [Bodirsky *et al.*, 2010a] Let $\theta(x_1, \ldots, x_n)$ be a formula in conjunctive normal form. We call θ reduced if it is not logically equivalent to any of its subformulas, i.e. there is no formula ψ obtained from θ by deleting literals of clauses such that $\theta(a) = \psi(a)$ for all $a \in \mathbb{Z}^n$.

An important property of reduced formulas is that if R is defined by a reduced formula $l_1 \vee \ldots \vee l_n$, then for each l_i, we can find a vector $\bar{x}$ that satisfies l_i but not l_j for all $j \neq i$.

Theorem 8 CSP($\Lambda_{\mathbb{Z}}$) is maximally tractable in CSP($\mathcal{H}_{\mathbb{Z}}$).

Proof: Let R be an arbitrary relation (of arity n) in $\mathcal{H}_{\mathbb{Z}}$ that does not satisfy requirement 2). Hence, there exists a real n-vector $\bar{y}$ and an infinite set $S \subseteq \mathbb{R}$ such that $\bar{y}$ satisfies R but for every $s \in S$, $s\bar{y}$ does not satisfy R. Assume without loss of generality that R is defined by a reduced formula $l_1(\bar{x}) \vee \ldots \vee l_k(\bar{x})$ where $l_1, \ldots, l_k$ are linear expressions.

Suppose that some $l_i \equiv p(\bar{x}) \neq c$ where $c \neq 0$. If $p(\bar{y}) \neq c$, then $p(k\bar{y}) \neq c$ for all $k \in \mathbb{R}^+$ except at most one, and the same holds for $R(k\bar{y})$. If $p(\bar{y}) = c$, then $p(k\bar{y}) \neq c$ for all $k \in \mathbb{R}^+$ except at most one, and the same holds for $R(k\bar{y})$. This leads to a contradiction and we can assume that if a literal $l_i \equiv p(\bar{x}) \neq c$, then $c = 0$.

If $\bar{y}$ satisfies some literal $l_i \equiv p(\bar{x}) \neq 0$, then $p(k\bar{y}) \neq 0$ for all $k \in \mathbb{R}$ except at most one, and the same holds for $R(k\bar{y})$. Thus, $\bar{y}$ only satisfies a literal $l_j \equiv q(\bar{x})ra$ where $r \in \{<, \leq, =, \geq, >\}$. By observing that $p(\bar{x}) < a \Leftrightarrow p(\bar{x}) \leq a - 1$, we may additionally assume that $r \in \{\leq, =, \geq\}$. Assume without loss of generality that $a \geq 0$; if $a < 0$, then consider the equivalent inequality obtained by multiplying with -1. If $r = (\geq)$, then $k\bar{y}$ satisfies R for all $k \geq 1$. Thus, $r \in \{\leq, =\}$. If $p(\bar{y}) = 0$, then $k\bar{y}$ satisfies R for all $k \in \mathbb{R}$ so we can safely assume that $a > 0$. We conclude that R is on one of the following forms: (1) $p(\bar{x}) = a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$ or (2) $p(\bar{x}) \leq a \vee q_1 \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0$ where $a > 0$.

Assume first that R is of type (1). In $\Lambda_{\mathbb{Z}} \cup \{R\}$, we can pp-define the following relation:

$$S(z) = \exists \bar{x}.(p(\bar{x}) = a \vee q_1(\bar{x}) \neq 0 \vee \ldots \vee q_n(\bar{x}) \neq 0) \wedge q_1(\bar{x}) = 0 \wedge \ldots \wedge q_n(\bar{x}) = 0 \wedge p(\bar{x}) = z.$$

The definition of R is reduced so there exists a vector $\bar{x}$ such that $p(\bar{x}) = a$ and $q_i(\bar{x}) = 0$, $1 \leq i \leq n$. Thus, $S(z)$ holds if and only if $z = a$; in other words, we have defined a non-zero constant. This implies that we can pp-define the constant 1 since $z = 1 \Leftrightarrow \exists x_1, ..., x_a, y. z = x_1 \wedge S(y) \wedge x_1 \geq 1 \wedge \ldots \wedge x_a \geq 1 \wedge y = x_1 + \ldots + x_a$. It is now straightforward to pp-define the relation $T_{0,1}$ and we have proved that CSP($\Gamma \cup \{R\}$) is locally NP-hard.

Assume instead that R is of type (2). Analogously to the construction of S, we can construct a non-empty unary relation S' that is upper bounded by a. Thus, S' contains a largest element b and the constant b can be pp-defined since $z = b \Leftrightarrow S(z) \wedge z \geq b$. The proof procedes as above. $\square$

The maximality proof also gives a characterisation of the relations $R \in \mathcal{H}_{\mathbb{Z}}$ that do not satisfy requirement 2): given a relation R, reduce it and check whether it is of one of the two 'bad' types identified in the proof. The maximality proof can also be generalised to a hardness result for constraint languages that are not necessarily subsets of $\mathcal{H}_{\mathbb{Z}}$.

Corollary 9 Let Γ be a constraint language over $\mathbb{Z}$ such that the relations $x = y + z$ and $x \geq 1$ are in $\langle\Gamma\rangle$. Then, $\Gamma \cup \{R\}$ is NP-hard whenever $R \in \mathcal{H}_{\mathbb{Z}} \setminus \Lambda_{\mathbb{Z}}$.

Proof: *(Sketch)* By Theorem 2, we may without loss of generality assume that $x = y + z$ and $x \geq 1$ are members of Γ. By inspecting the proof of Theorem 8, we see that CSP($\{R\} \cup X$) is NP-hard where X is a finite set of homogeneous equations and (possibly) a relation $p(\bar{x}) \geq a$ with $a > 0$. Every homogeneous equation can be pp-defined with the aid of $x = y + z$. Every relation $x \geq a$ with $a > 0$ can be pp-defined since $x \geq a \Leftrightarrow \exists y_1, \ldots, y_a . x = y_1 + \ldots + y_a \wedge y_1 \geq 1 \wedge \ldots \wedge y_a \geq 1$ and the equation $x = y_1 + \ldots + y_a$ is homogeneous. Thus, $p(\bar{x}) \geq a$ can be pp-defined since $p(\bar{x}) \geq a \Leftrightarrow \exists y . y = p(\bar{x}) \wedge y \geq a$. $\square$

3 Linear equations

In the previous section, we found a large maximally tractable subset $\Lambda_{\mathbb{Z}}$ of $\mathcal{H}_{\mathbb{Z}}$. Clearly, $\Lambda_{\mathbb{Z}}$ does not contain any linear equations $p(\bar{x}) = a$ with $a \neq 0$. We will now look at fragments of $\mathcal{H}_{\mathbb{Z}}$ that contain such equations. Similar fragments have been considered before: it is known that finding integer solutions to linear equation systems is a tractable problem [Kannan and Bachem, 1979]. A related problem has also been discussed in [Bodirsky *et al.*, 2010b, Section 6] but they restrict themselves to homogeneous equations. We will now work 'backwards' compared to the previous section; instead of starting with $\mathcal{H}_{\mathbb{Z}}$ and remove relations, we will extend the set of linear equations.

The algorithmic part will use results from [Cohen *et al.*, 2000] and a property known as *1-independence*. We note that the original definitions by Cohen et al. are slightly more general than those presented here; they do not restrict themselves to constraint languages. By the notation CSP$_{\Delta \leq k}(\Gamma \cup \Delta)$, we mean the CSP problem with constraints over $\Gamma \cup \Delta$ but where the number of constraints over Δ is less than or equal to k.

Definition 10 For two constraint languages Γ and Δ, we say that Δ is *k-independent with respect to* Γ if the following condition holds: any instance I of CSP($\Gamma \cup \Delta$) has a solution provided every subinstance of I belonging to CSP$_{\Delta \leq k}(\Gamma \cup \Delta)$ has a solution.

1-independence gives us a way to handle disjunctions efficiently. For constraint languages Γ and Δ, let the constraint language $\Gamma \between \Delta^*$ contain all relations $R(\bar{x}) \equiv c(\bar{x}) \vee d_1(\bar{x}) \vee \ldots \vee d_n(\bar{x})$, $n \geq 0$, where $c(\bar{x})$ is a constraint over Γ and $d_1(\bar{x}), \ldots, d_n(\bar{x})$ are constraints over Δ.

Theorem 11 [Cohen *et al.*, 2000] Let Γ and Δ be constraint languages. If CSP$_{\Delta \leq 1}(\Gamma \cup \Delta)$ is globally tractable, and Δ is 1-independent with respect to Γ, then CSP($\Gamma \between \Delta^*$) is globally tractable.

Let $\Gamma \subseteq \mathcal{H}_{\mathbb{Z}}$ denote all relations $p(\bar{x}) = b$ and $\Delta \subseteq \mathcal{H}_{\mathbb{Z}}$ denote all relations $p(\bar{x}) \neq b$. We will now prove that CSP($\Gamma \between \Delta^*$) is globally tractable (Theorem 12) and that it is a maximal tractable fragment of $\mathcal{H}_{\mathbb{Z}}$ (Theorem 13). We will also extend the maximality result in a way similar to Corollary 9; the omitted proof is analogous.

Theorem 12 CSP($\Gamma \between \Delta^*$) is globally tractable.

Proof: We first prove that Δ is 1-independent with respect to Γ. Let I_Γ be an instance of CSP(Γ) and I_Δ an instance of Δ. Assume that $I_\Gamma \cup \{d_i\}$ is satisfiable for each $d_i \in I_\Delta$ with $d_i \equiv p_i(\bar{x}) \neq c_i$.

We will perform an induction on the size of I_Δ. If $|I_\Delta| = 1$, then satisfiability follows from the assumptions. Assume that $|I_\Delta| = d$, $d > 1$, and that the statement holds for all $I'_\Delta \subset I_\Delta$. We show that $I_\Gamma \cup I_\Delta$ is satisfiable, too.

Let $I^i_\Delta = I_\Delta \setminus \{p_i(\bar{x}) \neq c_i\}$ and consider the instance $I_\Gamma \cup I^i_\Delta$ for each i. Let D_i, $1 \leq i \leq d$, be the set of satisfying points to these subproblems. The sets $D_1, \ldots, D_d$ are nonempty due to the induction hypothesis. Arbitrarily choose an element $\bar{x}_i \in D_i$ for each i. If $\bar{x}_i \in D_j$ for any $j \neq i$, then it is a solution to the entire instance and we are done. So we can assume that $p_i(\bar{x}_i) = c_i$ for all i.

Take two points $\bar{x}_1$ and $\bar{x}_2$ and define $\bar{x}^k = k\bar{x}_1 + (1-k)\bar{x}_2$ for $k \in \mathbb{Z}$. Observe that $\bar{x}^k$ satisfies I_Γ for all k. We will now show that there is a k such that $\bar{x}^k \in D_i$ for all i; by the previous comment, it is sufficient to consider the disequations.

For $i = 1$ we note that $p_1(\bar{x}^k) \neq c_1 \Leftrightarrow kp_1(\bar{x}_1) + (1-k)p_1(\bar{x}_2) \neq c_1 \Leftrightarrow (1-k)(p_1(\bar{x}_2) - c_1) \neq 0$ and since $p(\bar{x}_2) \neq c_1$ this is true for all $k \neq 1$. In the same way, we see that $\bar{x}^k \in D_2$ when $k \neq 0$.

For $i \neq 1, 2$, we note that if $p_i(\bar{x}_1) = d_1 \neq c_i$ and $p_i(\bar{x}_2) = d_2 \neq c_i$, then $p_i(\bar{x}^k) = k(d_1 - d_2) + d_2$. If $d_1 = d_2$, then the disequation is always true; otherwise, there is at most one value for k such that $p_i(\bar{x}^k) = c_i$. Hence, each disequation is not satisfied by $\bar{x}^k$ for at most one value of k, and we conclude that there is some k' such that $\bar{x}^{k'} \in D_i$ for all i. It follows that $I_\Gamma \cup I_\Delta$ is satisfiable for any size of I_Δ.

By Theorem 11, it is now sufficient to prove that CSP$_{\Delta \leq 1}(\Gamma \cup \Delta)$ is tractable. Let I be an instance of CSP$_{\Delta \leq 1}(\Gamma \cup \Delta)$. We view I as an equation system $A\bar{x} = \bar{b}$ together with a disequation $p(\bar{x}) \neq c$. We start by finding a satisfying integer point $\bar{x}$ to $A\bar{x} = \bar{b}$; this is tractable by [Kannan and Bachem, 1979]. If no such point exists, then I is not satisfiable. If the found solution $\bar{x}$ also satisfies $p(\bar{x}) \neq c$, then we have found a solution to I, too. Otherwise, note that if $\bar{y} \neq \bar{x}$ and $A\bar{y} = \bar{b}$, then $A(\bar{y} - \bar{x}) = \bar{b} - \bar{b} = \bar{0}$. By letting $\bar{x}_h = \bar{y} - \bar{x}$, we see that any satisfying point $\bar{z}$ can be written as $\bar{z} = \bar{x} + \bar{x}_h$ for some $\bar{x}_h$ such that $A\bar{x}_h = \bar{0}$. Since $p(\bar{x}) = c$, we note that $p(\bar{z}) \neq c \Leftrightarrow p(\bar{x}) + p(\bar{x}_h) \neq c \Leftrightarrow p(\bar{x}_h) \neq 0$. From this we conclude that we can find a solution to I if and only if we can find a point $\bar{x}_h$ such that $A\bar{x}_h = \bar{0}$ and $p(\bar{x}_h) \neq 0$.

Now we solve the system $A\bar{x} = \bar{0} \wedge p(\bar{x}) = 1$ over the rational numbers. If this system has no solution, then there is no point $\bar{x}_h$ since some rational multiple of $\bar{x}_h$ would have been a solution. If we find a solution $\bar{x}_q$ to this system, then there exists an integer $k \neq 0$ such that $k\bar{x}_q$ is an integer point satisfying $Ak\bar{x}_q = \bar{0}$ and $p(k\bar{x}_q) = k \neq 0$. We see that we can let $\bar{x}_h = k\bar{x}_q$ and conclude that I is satisfiable. As this only requires solving two linear systems, one over the integers and one over the rational numbers, this is a polynomial-time algorithm for solving CSP$_{\Delta \leq 1}(\Gamma \cup \Delta)$. $\square$

Theorem 13 $\Gamma \between \Delta^*$ is maximally tractable in $\mathcal{H}_{\mathbb{Z}}$.

Proof: The relations in $\mathcal{H}_{\mathbb{Z}} \setminus (\Gamma \between \Delta^*)$ are of the form $R \equiv p(\bar{x}) \leq c \vee \bigvee_{i=1}^{n} (q_i(\bar{x}) \neq a_i)$. Note that we do not have to consider relations with $<$ separately since those are always equivalent to a relation using $\leq$. We assume without loss of generality that the definition of R is reduced.

We will now show how to pp-define T_{z_0,z_1} for some $z_0 \neq z_1$ in $\mathbb{Z}$. By reasoning as in the proof of Theorem 8, we see that we can pp-define a unary relation $S(z)$ that is a subset of $\{z \in \mathbb{Z} \mid z \leq c\}$ by $S(z) \equiv \exists \bar{x}.(z = p(\bar{x})) \wedge (p(\bar{x}) \leq c \vee \bigvee_{i=1}^{n}(q_i(\bar{x}) \neq a_i)) \wedge (\bigwedge_{i=1}^{n}(q_i(\bar{x}) = a_i))$.

We first prove that $|S| > 1$. If $|S| = 0$, then $(\bigwedge_{i=1}^{n}(q_i(\bar{x}) = a_i)) \Rightarrow p(\bar{x}) > c$ but since the definition of R is reduced, we know that there exists an integral vector $\bar{x}$ satisfying $p(\bar{x}) \leq c$. If $|S| = 1$, then $\bigwedge_{i=1}^{n}(q_i(\bar{x}) = a_i)) \Rightarrow p(\bar{x}) = d$ for some $d \leq c$ but then $R = \mathbb{Z}^n$ and we have a contradiction since $R \in \Gamma \curlyvee \Delta^*$. To see this, let $\bar{x}$ be an arbitrary vector in $\mathbb{Z}^n$. If it satisfies one of the disequations, then we are done. Otherwise, $p(\bar{x}) = d$ and it clearly satisfies the inequality $p(\bar{x}) \leq c$. Consequently, $|S| > 1$.

Define $z_0 = \max\{z \mid S(z)\}$ and $z_1 = \max\{z \mid S(z), z \neq z_0\}$ and note that these constants are pp-definable in $\Gamma \cup \{R\}$. Now, $T_{z_0,z_1}(x, y, z) \equiv S(x) \wedge S(y) \wedge S(z) \wedge x + y + z = (2z_0 + z_1)$ and NP-hardness follows from Theorem 2. $\square$

Corollary 14 Let Γ be a constraint language over $\mathbb{Z}$ such that the relations $x = y + z$ and $x = 1$ are in $\langle\Gamma\rangle$. Then, $\Gamma \cup \{R\}$ is NP-hard whenever $R \in \mathcal{H}_{\mathbb{Z}} \setminus (\Gamma \curlyvee \Delta^*)$.

4 Constraints that are k-valid

We will now demonstrate that there are an infinite number of distinct maximally tractable fragments within $\mathcal{H}_{\mathbb{Z}}$. This fact makes complexity classifications harder since a description of the tractable cases must be more elaborate than just listing the maximally tractable fragments.

A relation R is said to be *k-valid*, $k \in \mathbb{Z}$, if $(k, \ldots, k) \in R$. A constraint language Γ is k-valid if every relation in Γ is k-valid. Let Γ_k, $k \in \mathbb{Z}$, denote the set of k-valid relations in $\mathcal{H}_{\mathbb{Z}}$ together with the empty relation. Clearly, $\Gamma_i \neq \Gamma_j$ whenever $i \neq j$; Γ_i contains the relation $(x = i)$ but does not contain $(x = j)$ and vice versa. Solving instances of CSP(Γ_k) is obviously trivial (simply check whether some constraint is based on the empty relation) but such classes have to be considered, too, in order to obtain full complexity classifications. The maximality proof for k-valid constraints differs slightly from the proofs in the preceeding sections. There, we managed to construct explicit NP-hard constraint languages. This proof is slightly non-constructive since we obtain a sequence of constraint languages and prove that (at least) one of them is NP-hard. However, we do not know which one.

For distinct $a, b, c \in \mathbb{Z}$, define $T'_{a,b,c}(x, y) \equiv \{a, b, c\}^2 \setminus \{(a, a), (b, b), (c, c)\}$. CSP($\{T'_{a,b,c}\}$) is an NP-hard problem since it corresponds to the 3-COLOURABILITY problem.

Theorem 15 For each $k \in \mathbb{Z}$, Γ_k is a maximally tractable language in $\mathcal{H}_{\mathbb{Z}}$.

Proof: The problem CSP(Γ_k) is obviously globally tractable. To prove maximality, arbitrarily choose a relation $R \in \mathcal{H}_{\mathbb{Z}}$ that is not k-valid. Let m denote the arity of R and consider the following relations:

$$U_1(z) \equiv \exists y, x_2, \ldots, x_m.R(z, x_2, x_3, \ldots, x_m) \wedge y = k$$
$$U_2(z) \equiv \exists y, x_3, \ldots, x_m.R(y, z, x_3, x_4, \ldots, x_m) \wedge y = k$$
$$\vdots$$
$$U_m(z) \equiv \exists y.R(y, y, y, \ldots, y, z) \wedge y = k$$
$$U_{m+1}(z) \equiv \exists y.R(y, y, y, \ldots, y, y) \wedge y = k$$

The relations $U_1, \ldots, U_{m+1}$ are pp-definable in $\Gamma_k \cup \{R\}$ since the relation $y = k$ is k-valid. We claim that there exists an index $1 \leq j \leq m$ such that $U_j \neq \varnothing$ and $k \notin U_j$. Since R is not k-valid, it follows that $U_{m+1} = \varnothing$ so there exists a smallest index $2 \leq j' \leq m + 1$ such that $U_{j'} = \varnothing$. Let $j = j' - 1$. Clearly, U_j is non-empty and if $k \in U_j$, then $U_{j+1} = U_{j'}$ is non-empty which leads to a contradiction.

We now let $c_a(z) \equiv (z = a)$ and pp-define the relation $c_{k'}(z)$ for some $k' \neq k$. Assume without loss of generality that there is some element in U_j that is larger than k; if not, then there is some element in U_j that is smaller than k and the reasoning is symmetric. Let $k' = \min\{x \in U_j \mid x > k\}$ and note that $z = k' \Leftrightarrow U_j(z) \wedge (z \geq k) \wedge (z \leq k')$. The relations $(z \geq k)$ and $(z \leq k')$ are both k-valid so $(z = k)$ is pp-definable in $\Gamma_k \cup \{R\}$. Using the relation $z = k'$, we conclude the proof by the following pp-definition: $T'_{k-1,k,k+1}(x, y) \equiv \exists z, w.(z = w \vee x \neq y) \wedge c_k(z) \wedge c_{k'}(w) \wedge \wedge(k - 1 \leq x) \wedge (x \leq k + 1) \wedge (k - 1 \leq y) \wedge (y \leq k + 1)$. NP-hardness of CSP($\{T'_{k-1,k,k+1}\}$) implies NP-hardness of CSP($\Gamma_k \cup \{R\}$) via Theorem 2. $\square$

5 Discussion

The results reported in this paper constitute a step towards a better understanding of the complexity of temporal reasoning in discrete time structures. Below, we discuss how this work can be continued.

5.1 Horn DLRs

Completely classifying the complexity of CSP($\mathcal{H}_{\mathbb{Z}}$) appears to be a quite hard problem. Consider the NP-complete integer feasibility problem: given a system of inequalities $Ax \geq b$, decide whether there exists a satisfying integer vector x or not. Note that each row of the system can be viewed as a relation in $\mathcal{H}_{\mathbb{Z}}$. Thus, a complete classification of CSP($\mathcal{H}_{\mathbb{Z}}$) would give us a classification of the integer feasibility problem (parameterised by allowed row vectors). Such a classification is not currently known and, in fact, there are no classifications even if we restrict ourselves to finite domains or if we consider the closely related integer optimisation problem.

One obvious difficulty when classifying CSP($\mathcal{H}_{\mathbb{Z}}$) is that we do not know what algorithmic techniques will be needed. The results in this paper are based on either solving linear equations or solving linear programming problems over the real numbers. Completely different methods may be needed in other cases, though. As a concrete example, consider the constraint language Γ containing all relations of the type $ax + by = c$, $x \leq c$, and $x \geq c$ (where $a, b, c \in \mathbb{Z}$) and note that $\Gamma \subseteq \mathcal{H}_{\mathbb{Z}}$. Bodirsky et al. [2009] have shown that CSP(Γ) is tractable by using a graph-theoretic approach; it is not clear how (or if) this algorithm can be recast in more familiar terms.

Another difficulty is that there are tractable cases where we have not been able to prove maximality. One example is the previously mentioned class by Bodirsky et al. Another example is constraints of the types $ax - by \leq c$ and $ax - by = c$ where $a, b \in \{0, 1\}$ and $c \in \mathbb{Z}$. Let $\Sigma_{\mathbb{Z}}$ denote the corresponding constraint language. The tractability of CSP($\Sigma_{\mathbb{Z}}$) problem follows from combining Theorem 19.1 and Theorem 19.3(iv) in [Schrijver, 1986] with the fact that linear programs can be solved in polynomial time. This result is interesting since it implies tractability of the discrete-time analogue of Dechter et. al's [1991] well-known *simple temporal networks*.

Deciding maximality for $\Sigma_{\mathbb{Z}}$ and similar classes appear to be non-trivial. Apparently, $\Sigma_{\mathbb{Z}}$ may be difficult to extend with disequality relations $p(\bar{x}) \neq a$. It is, for instance, straightforward to prove that CSP($\Sigma_{\mathbb{Z}} \cup \{x \neq y\}$) is NP-hard. It may be the case that $\Sigma_{\mathbb{Z}}$ can be extended in other ways, though.

5.2 Semilinear relations and DLRs

If we turn our attention to semilinear relations and DLRs, then we immediately note that they give rise to a much richer class of CSPs than Horn DLRs. The following is an interesting consequence: for every finite constraint language Γ over a finite domain D, there exists a finite set $\Gamma' \subseteq \mathcal{S}_{\mathbb{Z}}$ such that CSP(Γ) and CSP(Γ') are polynomial-time equivalent. This can be demonstrated as follows: given a relation $R \subseteq D^k$ where $D = \{d_1, \ldots, d_m\}$ is finite, define $R'(x_1, \ldots, x_k) \equiv \bigwedge_{i=1}^{k} (x_i = d_1 \vee \ldots \vee x_i = d_m) \wedge \bigwedge_{(t_1,\ldots,t_k) \in D^k, (t_1,\ldots,t_k) \notin R} (x_1 \neq t_1 \vee \ldots \vee x_k \neq t_k)$. It is now straightforward to see that R' is a semilinear relation and that CSP($\{R\}$) is polynomial-time equivalent to CSP($\{R'\}$). This idea is straightforward to extend to constraint languages, so a complete classification of CSP($\mathcal{S}_{\mathbb{Z}}$) would also constitute a complete classification of finite-domain CSPs. Such a classification has for many years been a major open question within the CSP community. We can also observe that the resulting constraint language Γ' is typically not a subset of $\mathcal{D}_{\mathbb{Z}}$. The simpler structure of $\mathcal{D}_{\mathbb{Z}}$ may very well simplify the classification task. We note that the finite-domain CSP problem for so-called *clausal* relations is completely classified [Creignou *et al.*, 2008], and this kind of relations are defined by single clauses satisfying certain properties.

When studying the complexity of CSP($\mathcal{S}_{\mathbb{Z}}$) and CSP($\mathcal{D}_{\mathbb{Z}}$), it appears that the known sources of tractable fragments increase drastically. An obvious example are the relations that are first-order definable over $(\mathbb{Q}; <)$. Every tractable subclass has been identified by Bodirsky and Kara [2010] so we let Γ denote one of their tractable classes. The structure $(\mathbb{Q}; <)$ admits quantifier elimination and $\Gamma \subseteq \mathcal{S}_{\mathbb{Z}}$. By combining this fact with Lemma 4, it can be shown that an instance I of CSP(Γ) has a solution if and only if $I|_{\mathbb{Z}}$ has a solution. Thus, CSP($\Gamma|_{\mathbb{Z}}$) is tractable, too. Note that there is no a priori reason to believe that $\Gamma|_{\mathbb{Z}}$ is maximally tractable within $\mathcal{D}_{\mathbb{Z}}$ or $\mathcal{S}_{\mathbb{Z}}$ (even though Γ is maximal within the relations that are first-order definable within $(\mathbb{Q}, <)$). It may be the case that there are many different tractable classes that contains $\Gamma|_{\mathbb{Z}}$.

References

[Angluin *et al.*, 2007] D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distrib. Comput.*, 20:279–304, 2007.

[Bettini *et al.*, 1998] C. Bettini, X. Wang, and S. Jajodia. A general framework for time granularity and its application to temporal reasoning. *Ann. Math. Artif. Intell.*, 22(1–2):29–58, 1998.

[Bodirsky and Kára, 2010] M. Bodirsky and J. Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.

[Bodirsky *et al.*, 2009] M. Bodirsky, G. Nordh, and T. von Oertzen. Integer programming with 2-variable equations and 1-variable inequalities. *Inform. Process. Lett.*, 109(11):572–575, 2009.

[Bodirsky *et al.*, 2010a] M. Bodirsky, H. Chen, and M. Pinsker. The reducts of equality up to primitive positive definability. *J. Symb. Log.*, 75(4):1249–1292, 2010.

[Bodirsky *et al.*, 2010b] M. Bodirsky, P. Jonsson, and T. von Oertzen. Horn versus full first-order: complexity dichotomies in algebraic constraint satisfaction. arXiv:1005.1141v2.

[Boujjani and Habermehl, 1996] A. Boujjani and P. Habermehl. Constrained properties, semilinear systems, and Petri nets. In *Proc. of CONCUR '96, Concurrency Theory, 7th International Conference*, pages 481–497, 1996.

[Cohen *et al.*, 2000] D. Cohen, P. Jeavons, P. Jonsson, and M. Koubarakis. Building tractable disjunctive constraints. *J. ACM*, 47:826–853, September 2000.

[Creignou *et al.*, 2008] N. Creignou, M. Hermann, A. Krokhin, and G. Salzer. Complexity of clausal constraints over chains. *Theor. Comput. Syst.*, 42(2):239–255, 2008.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artif. Intell.*, 49:61–95, 1991.

[Fisher *et al.*, 2005] M. Fisher, D. Gabbay, and L. Vila, editors. *Handbook on Temporal Reasoning in Artificial Intelligence*. Elsevier, 2005.

[Jeavons, 1998] P. Jeavons. On the algebraic structure of combinatorial problems. *Theoret. Comput. Sci.*, 200(1–2):185–204, 1998.

[Jonsson and Bäckström, 1998] P. Jonsson and C. Bäckström. A unifying approach to temporal constraint reasoning. *Artif. Intell.*, 102(1):143–155, 1998.

[Kannan and Bachem, 1979] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.*, 8(4):499–507, 1979.

[Krokhin *et al.*, 2004] A. Krokhin, P. Jeavons, and P. Jonsson. Constraint satisfaction problems on intervals and lengths. *SIAM J. Discrete Math.*, 17(3):453–477, 2004.

[Meiri, 1996] I. Meiri. Combining qualitative and quantitative constraints in temporal reasoning. *Artif. Intell.*, 87(1-2):343–385, 1996.

[Parikh, 1966] R. Parikh. On context-free lanugages. *J. ACM*, 13(4):570–581, 1966.

[Schrijver, 1986] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1986.